DOI 10.18.276/978-83-8419-053-1-32

28th European Conference on Artificial Intelligence (ECAI 2025)

SPRINT INTO THE FUTURE: INTELLIGENT MANAGEMENT THROUGH LLM-POWERED COMPUTER SIMULATIONS IN AGILE PROJECTS

Tomasz Wiśniewski[a], Rafał Szymański[b], Marta Starostka-Patyk[c]

[a] *University of Szczecin, Institute of Management, Poland*
[b] *University of Szczecin, Institute of Spatial Management and Socio-Economic Geography Poland*
[c] *University of Technology Częstochowa, Częstochowa, Poland*

ABSTRACT

Purpose: *This study investigates how Large Language Models (LLMs) integrated with computer simulations can enhance intelligent management in Agile Project Management. It examines generative AI's ability to create simulation models from natural language, enabling faster and more accurate scenario analysis. The research integrates Agile Earned Value Management (AgileEVM) metrics to support decision-making, optimize resources, and improve adaptability in complex IT projects.*

Need for the study: *Traditional project management tools require specialized skills and struggle with evolving Agile requirements. As IT projects grow in complexity, solutions that merge technical precision with operational flexibility are essential. LLM-generated simulations provide advanced analytics without coding expertise, addressing gaps in speed, accessibility, and adaptability.*

Methodology: *A framework using GPT-4 generated Python-based simulations from textual project descriptions, incorporating AgileEVM metrics (CPI, SPI, velocity, defect rates). A case study tested three project scenarios with varying cohesion, leadership, and complexity. GPT-4 outputs were benchmarked against ARENA discrete-event simulation and Asana AI across 20 sprints.*

Findings: *GPT-4 simulations matched ARENA results with <2% CPI/SPI variation and <10% defect metric deviation. Team cohesion, leadership quality, and complexity strongly affected Agile performance.*

Practical Implications: *The approach allows managers to build high-fidelity simulation tools rapidly, reducing time and cost. Integrating AgileEVM metrics enables "what-if" analyses to anticipate risks and refine strategies, promoting data-driven, intelligent management in Agile environments.*

Keywords: Large Language Models; Agile Project Management; Computer Simulation; IT Project Planning; AI-Driven Decision Making

Jel codes: C63; C88;

1. INTRODUCTION

IT projects play a crucial role in driving technological innovation and transforming various sectors of the economy, including healthcare, finance, education, and logistics. They serve as catalysts for change, offering novel solutions to traditional challenges and shaping modern business strategies (Lindsjørn et al., 2016; Pricopoaia et al., 2024). The final selection of an optimal project development approach should be based on theoretically sound criteria, account for specific user needs and project requirements, and consider key factors such as team size and expertise, project budget and deadlines, and task complexity (Schwaber & Sutherland, 2020). Therefore, researching software development processes and methodologies is crucial for understanding industry trends and identifying the best strategies for sustainable business growth. This raises the need to adapt existing tools developed for project management so that they can also be used in projects conducted according to agile methodologies and aligned with the principles of intelligent management.

Thus, one of the key challenges in project management is modeling development alternatives and determining the most effective approach, which serves as the primary objective of this study. The framework proposed in this article allows the integration of Large Language Models (LLMs) with computer simulation using the AgileEVM methodology. The use of this concept will provide the ability to strategically plan approaches to selecting quality projects and enable project managers to radically facilitate and accelerate such planning.

2. THEORETICAL ASPECTS OF PROJECT MANAGEMENT AND PERFORMANCE EVALUATION

Project management is an interdisciplinary field encompassing the planning, organization, execution, and control of activities to achieve predefined objectives within specified time, budget, and quality constraints. A defining characteristic of projects is their temporary nature, well-defined scope, and the necessity for efficient resource utilization (Project Management Institute, 2018). In contemporary contexts, these practices are increasingly being shaped by the principles of intelligent management, which integrate data-driven decision-making, advanced analytics, and adaptive strategies. Modern project management approaches (Dingsøyr et al., 2012; Schwaber & Sutherland, 2020) have evolved from traditional waterfall methodologies to Agile frameworks, emphasizing flexibility, iteration, and adaptability to dynamic market and technological conditions. Agile methodologies encompass several distinct frameworks, each tailored to different project needs, scope management approaches, and team interactions:

1. Scrum – One of the most widely adopted Agile methodologies, Scrum follows an iterative approach in which work is divided into short cycles called Sprints. It defines specific roles (Product Owner, Scrum Master, Development Team) and ceremonies, such as Daily Scrum, Sprint Review, and Sprint Retrospective, to ensure transparency and continuous improvement (Schwaber & Sutherland, 2020).
2. eXtreme Programming (XP) – A software development methodology that prioritizes code quality and frequent client interactions. XP employs practices such as pair programming, continuous integration, and unit testing to minimize errors through iterative development and close developer collaboration (Beck, 2005).
3. Crystal Family – A collection of lightweight Agile methodologies designed to accommodate teams of different sizes and characteristics. Crystal methodologies emphasize direct communication, adaptability, and process flexibility, enabling teams to tailor their workflows to project-specific requirements (Cockburn, 2004).
4. Adaptive Software Development (ASD) – A framework centered on dynamically adjusting project strategies in response to evolving business and technological conditions. ASD promotes iterative software delivery, continuous experimentation, and adaptive learning to respond to uncertainties effectively (Highsmith, 2007).
5. Dynamic Systems Development Method (DSDM) – A structured Agile framework primarily applied in large-scale business projects. DSDM maintains rigorous control over project scope and requirements while allowing flexibility in prioritizing deliverables. It requires continuous stakeholder involvement and iterative product development cycles (Stapleton, 1997).

6. Feature-Driven Development (FDD) – An Agile approach focusing on systematically delivering functional features aligned with business priorities. FDD emphasizes domain modeling, feature-based planning, and iterative deployment of completed system components (Palmer & Felsing, 2002).

One of the most critical aspects of project management is evaluating project performance, which refers to the ability to achieve predetermined objectives optimally in terms of time, cost, and quality. In Agile methodologies, specific performance metrics provide both quantitative and qualitative insights into team productivity and product quality. Velocity and delivery speed are among the key indicators—velocity refers to the number of work units (e.g., story points, completed tasks) delivered by a team in a single iteration (Sprint). This metric enables forecasting future work pace and iteration planning. Delivery speed, on the other hand, measures the time elapsed between task initiation and final implementation. High-performing teams aim to minimize delivery speed by eliminating inefficiencies and operational bottlenecks. Another important factor is team stability, as team composition variability affects efficiency and long-term project success. Team stability is measured by analyzing turnover rates and the number of new members joining within a given period. A stable team fosters synergy, improves communication, and enhances coordination, all of which contribute to increased performance strategies (Lindsjørn et al., 2016). Additionally, project success is not solely determined by speed but also by the quality of the deliverables. In IT projects, product quality is assessed using metrics such as defect rates, test coverage, failure rates, and compliance with user requirements. High product quality reduces rework costs and enhances stakeholder satisfaction (Boehm, B., & Turner, 2004). Finally, team efficiency evaluates a team's productivity in terms of delivering business value. It can be assessed by comparing planned versus completed work, resource utilization, and the level of collaboration and self-organization within the team. High-performing teams exhibit strong autonomy, effective communication, and the ability to quickly adapt to shifting project priorities (Dingsøyr et al., 2012). In conclusion, project management effectiveness depends on both the selected methodologies and a team's ability to manage resources efficiently, overcome challenges, and deliver high-quality outcomes. Metrics such as velocity, team stability, product quality, and team efficiency are essential indicators of project performance and enable organizations to optimize their workflows to maximize value.

3. ADVANCED ANALYSIS OF THE EARNED VALUE MANAGEMENT SYSTEM IN THE CONTEXT OF AGILE METHODOLOGIES

The Earned Value Management System (EVMS) constitutes a sophisticated project management mechanism that integrates key performance dimensions, including scope, schedule, and cost, to ensure precise planning, performance monitoring, and the implementation of control mechanisms. By leveraging data generated by EVMS, project management professionals can conduct advanced predictive analyses regarding costs, schedules, and technical project outcomes. The implementation of EVMS necessitates adherence to established industry standards (National Defense Industrial Association, 2018), which regulate the execution of various subprocesses, such as contract management, team organization, and risk monitoring.

A fundamental aspect of EVMS is the establishment of a realistic Performance Measurement Baseline (PMB), which serves as a reference point for the systematic evaluation of project status and facilitates informed managerial decision-making. From the perspective of project management sciences, the functionality of EVMS remains highly dependent on its socio-organizational environment, meaning that factors such as team competencies, decision-making processes, and stakeholder engagement may exert either a positive or negative influence on system effectiveness. In academic discourse, EVMS is classified as a socio-technical system, necessitating the simultaneous refinement of both its technological components and the social dynamics surrounding it (Cho et al., 2020).Within this framework, EVMS maturity is defined as the degree to which its subprocesses comply with established industry standards and their effectiveness in project execution. Studies have demonstrated that a higher level of EVMS maturity correlates with improved project performance, as a well-integrated EVMS enhances project control and predictive accuracy (Aramali et al. 2022; Government Accountability Office 2021). Enhancing the effectiveness of EVMS, and consequently increasing the probability of project success, therefore requires the optimization of both technical processes and the inclusion of social determinants in the assessment of its functionality (Cho et al., 2020). In the context of Agile methodologies, the implementation of EVMS necessitates the adaptation of traditional performance metrics to accommodate the iterative nature of project execution. The key performance indicators (KPI) employed in Agile EVM are presented in table 1.

Table 1. KPI in Agile EVM

Metric	Description
Planned Value (PV)	Represents the planned value of work to be completed within a given timeframe.
Earned Value (EV)	Reflects the actual value of work completed, in accordance with the defined project scope.
Actual Cost (AC)	Aggregates the financial expenditures incurred up to a specific point in the project lifecycle.
Cost Variance (CV)	Denotes the difference between EV and AC, indicating either budget efficiency or cost overruns.
Cost Performance Index (CPI)	The ratio of EV to AC, providing a consolidated measure of cost efficiency.
Schedule Variance (SV)	The difference between EV and PV, determining the extent to which the schedule adheres to planned objectives.
Schedule Performance Index (SPI)	The ratio of EV to PV, facilitating the assessment of project progress relative to the planned schedule.

Source: own elaboration based on (Aramali et al. 2022; Government Accountability Office 2021).

In Agile project management frameworks, such as Scrum, execution follows an iterative approach (Sprints), wherein work scope is defined through user stories and evaluated based on story points. The effective deployment of Agile EVM necessitates the completion of several key steps. First, it is essential to determine the total number of story points in a release, as the sum of all points assigned to user stories within a given release defines the Budget at Completion (BAC). Next, teams must monitor work progress by updating the number of completed story points at the conclusion of each iteration, which allows for the dynamic calculation of EV. Simultaneously, tracking actual costs through continuous monitoring of project expenditures ensures an accurate assessment of the Actual Cost (AC). Finally, performance metrics such as CV, CPI, SV and SPI should be calculated regularly to enable real-time project evaluation and support timely, data-driven corrective decision-making. The adoption of Agile EVM is associated with a range of organizational benefits. One key advantage is the early detection of deviations, as systematic monitoring of EVM metrics enables the prompt identification of schedule and budget discrepancies, allowing for timely mitigation. Additionally, the use of indicators such as the CPI and SPI enhances forecasting accuracy, supporting more precise predictions regarding costs and completion timelines, which in turn aids strategic decision-making. Agile EVM also contributes to increased transparency in project governance by providing stakeholders with objective, quantifiable data on project status, thereby fostering trust and improving communication among teams.

Agile EVM serves as an advanced mechanism that integrates classical earned value management techniques with flexible Agile project management methodologies. It enables precise assessment of project progress, resource utilization efficiency, and alignment with planning objectives. The successful implementation of Agile EVM is contingent on both its technical deployment and the ability to adapt to the socio-organizational environment of a project, necessitating an interdisciplinary approach to project management.

4. IMPLEMENTATIONS OF SIMULATION MODELLING AND AI IN PROJECT MANAGEMENT

4.1. Use of computer simulation in project management

Simulation modeling involves creating computer models that represent real systems or processes. It enables the replication of real-world operations using mathematical models, allowing for the analysis of how external variables (input factors) and internal parameters influence system behavior. Today, simulation modeling is widely utilized in both science and management. In the field of management, it plays a key role in cost reduction, accelerating development cycles, improving product and service quality, and enhancing knowledge management (Banks et al., 2005; Wooley et al., 2023). Simulation modelling has been a productive topic of research in many fields, including project management applications, particularly concerning IT projects (Rodríguez et al., 2011). Pikh et al. (2025) use simulation modeling to automate the procedure for selecting the optimal solution for the IT project development process, providing reliable support and increasing the efficiency of project management. Hanne and Nickel (2005) considered the problem of planning inspections and other tasks within a software development project with respect to the objectives of quality, project duration, and costs. They built a discrete-event simulation model comprising the phases of coding, inspection, test, and rework and formalised the problem of project scheduling as a multiobjective simulation optimisation problem. Authors in Ratushny et al. (2019) developed a computer model for estimating scenarios for fire system development projects. The model is based on an algorithm that simulates these projects, as well as a rapid assessment of each scenario for the development of fire protection system environments. To support decision-making in project management application of simulation modelling and statistical control charts can be used (Acebes et al. 2014). Beyond risk analysis, mitigating uncertainty is crucial for achieving project objectives. To address this, Naeni and Salehipour (2011) represented percent completions as fuzzy numbers and applied fuzzy set theory to estimate project performance.

4.2. Gen-AI in Project Management: support and automation

Advanced applications of Generative Artificial Intelligence (Gen-AI) gained widespread recognition in 2022 as user-friendly tools emerged, enabling automation and optimization in fields such as project management and engineering with unprecedented efficiency and adaptability (Durth et al., 2023). Gen-AI is a field of artificial intelligence that focuses on creating new content, such as text, images or program code, by using advanced machine learning models. In the context of large language models (LLMs) and natural language processing (NLP), gen-AI plays a key role in automating and improving natural language processes (AlShannaq et al., 2024). Large language models are sophisticated natural language processing algorithms that have been trained on huge text datasets to predict the next words in a sentence, translate text, answer questions or generate consistent and meaningful answers. Generative artificial intelligence has great potential in project management, streamlining key processes, increasing efficiency and supporting decision-making. With properly trained LLMs, Gen-AI can significantly improve project planning, monitoring and execution. A significant advancement in the application of artificial intelligence to content generation has been made possible by the release of apps like ChatGPT, Claude, Perplexity, which provide end users with access to LLMs (Uppalapati & Nag, 2024; Choudhary, 2025). In recent years, the most widely used Gen-AI model has been GPT-3 (later GPT-3.5 and GPT-4), particularly through its application as ChatGPT, which has gained significant traction not only in management but also in education, business, programming, and various other domains, contributing to its rapid rise in popularity (Mossavar-Rahmani & Zohuri, 2024). With such rapid development, artificial intelligence models are becoming a helpful tool in project management. One of the most significant applications of AI in this field is the automation and enhancement of project planning (Sravanthi, 2023). Artificial intelligence models can analyze historical data, identify patterns, and use them to predict realistic schedules, resource allocation, and potential risks. This enables managers to make faster and more accurate decisions regarding execution strategies. Gen-AI can also generate detailed project plans, taking into account client requirements, available resources, and operational risks (Crawford et al., 2023). AI models are capable of simulating various scenarios and assessing their impact on project execution, allowing for better preparation for potential threats (Auth et al., 2019). Another key functionality of AI in project management is the automation of reporting and progress monitoring

(Hofmann et al., 2020). These models can generate reports based on data from multiple sources, summarizing project status, identifying delays, and recommending corrective actions.

4.3. AI and Simulation-Based Project Management: A Structured Framework

The rapid expansion of modeling languages has reignited interest in the long-standing challenge of automating code generation and program synthesis. Tools like OpenAI's Codex and GitHub Copilot are reshaping software development by offering advanced automated coding capabilities. By comprehending both programming syntax and human-readable instructions, these tools help developers write, debug, and optimize code more efficiently. Codex and Copilot convert natural language prompts into functional code, significantly reducing the time required for routine tasks and boosting developer productivity (Chen et al., 2021). Beyond assisting programmers, these tools also broaden access to coding by lowering the barrier for non-programmers to create simulation models. By allowing users to interact using natural language, they make coding more intuitive for individuals without formal programming expertise (Nguyen et al., 2023). As demonstrated in Jackson's study (Jackson et al., 2023), automated code generation based on supply chain management problem descriptions can be a valuable tool for developing simulation models. As project management becomes increasingly complex, organizations are turning to advanced technologies to enhance decision-making, optimize workflows, and predict potential risks. Computer simulations and generative AI offer innovative solutions by modeling project scenarios, automating repetitive tasks, and improving resource allocation. Proposed framework (fig. 1) outlines the integration of these technologies into project management processes to increase efficiency and reduce uncertainties.

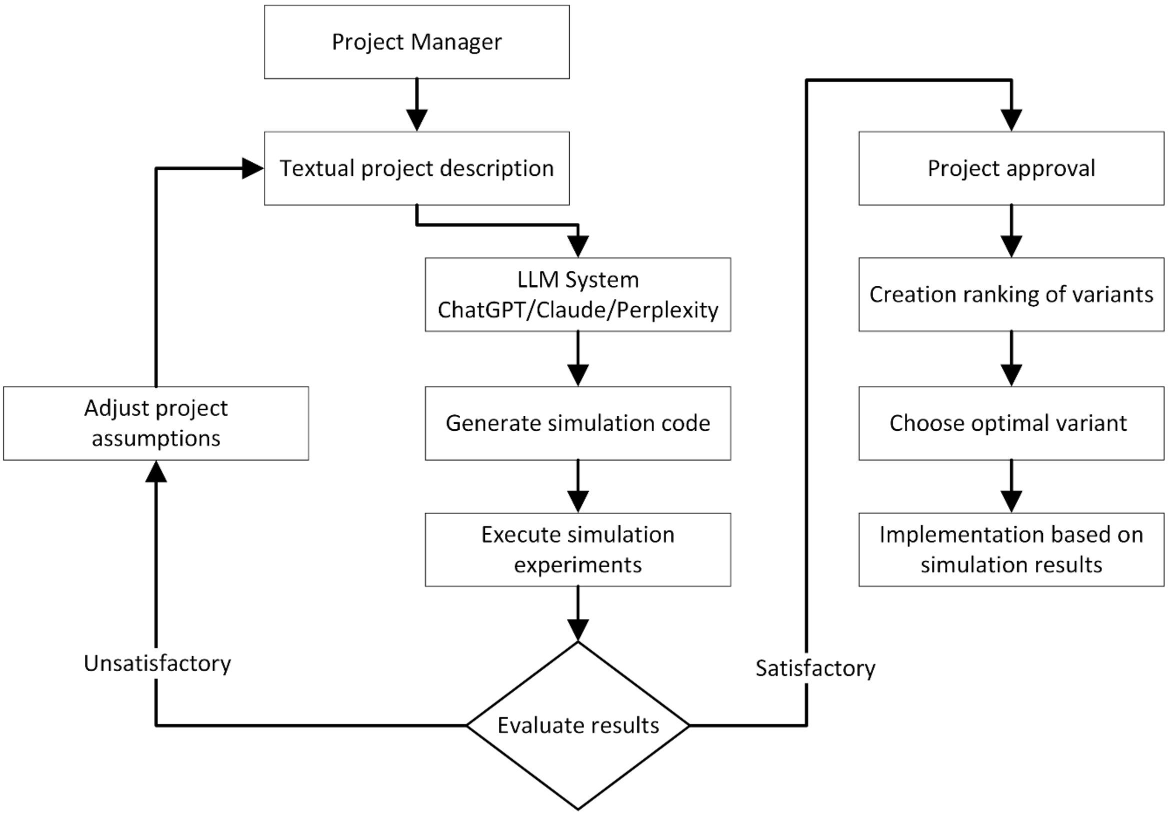

Figure 1. Framework for integrating computer simulation and Gen-AI in Project Management
Source: own elaboration.

The proposed framework focuses on integrating the capabilities of generative AI and simulation modeling. A project manager can create a simulation model (program code) by providing a textual description of the analyzed project, using tools such as ChatGPT, Claude, or Perplexity, which are based on LLMs. The generated simulation model code is then executed.

The results of these simulation experiments either lead to the approval of the project or require adjustments to the project assumptions or decisions, followed by a new series of simulation experiments. This approach allows for the preliminary selection of variants and the creation of a ranking. Among the variants that pass the initial selection, the optimal one is chosen for implementation, with the basis for evaluation being the results of simulation experiments for selected scenarios.

5. RESULTS

To show the practical application of the presented framework integrating computer simulation and Gen-AI in Project Management, we used the case-study presented in (Szyjewski & Muszyńska, 2013). It showed how to use simulation modeling with Earned Value in project management using the Agile approach. The case-study presented in our work was extended to 3 different scenarios (Table 2) to show the applicability of simulation in what-if analysis. To ensure comparability and clarity, the scenarios were defined using a consistent set of parameters covering project scale, team structure, leadership, complexity, and key Agile-based performance indicators such as velocity, CPI, and SPI. These parameters were selected based on their relevance in empirical studies on Agile performance, as well as their ability to capture both quantitative (e.g., cost, duration, delivery speed) and qualitative aspects (e.g., team cohesion, product quality) of project execution. This approach allowed us to simulate realistic variations in team dynamics, delivery outcomes, and management challenges across different project contexts.

Table 2. Simulation scenarios assumptions

Parameter	**Project 1**	**Project 2**	**Project 3**
Project Value [PLN]	500	2,000,000	1,200,000
Team Size	6	8	10
Team Composition	Domain experts with experience	Domain experts from different divisions	Mixed-experience, multi-role team
Team Cohesion	Cooperative, task-focused	Low, conflicts and blame-shifting	Moderate, improved over time
Complexity [Story Points]	10	20	15
Sprint Length [weeks]	2	2	2
Number of Sprints	20	20	20
Project Leader Soft Skills (0-1)	0.7	0.9	0.5
Error Rate [%]	5%	15%	8%
Bug Detection Efficiency [%]	90%	95%	88%
Delivery Delay [weeks]	2	6	3
Post-release Problems	Minor, non-critical bug fixes	High cost, low quality, low margin	Moderate quality issues
Average Velocity	~474	~437	~475
CPI (avg)	0.02	0.00	0.01
SPI (avg)	0.95	0.42	0.63

Source: own elaboration.

The approach we presented shows how simple simulation models can be created automatically from natural language verbal descriptions, and how project managers and AI-based systems can work together. Table 3 shows the prompt used in the ChatGPT4 application to generate simulation code.

Table 3. Sample prompt for simulation model in project management with Earned Value

Prompt for GPT application
Acting on the assumptions given in the table: “Table 2 added here” Provide Python code to simulate the described project management problem. Present the main results of the project evaluation based on the Earned Value methodology using charts, including CPI, SPI, velocity, number of defects, and average bug fix time

Source: own elaboration.

It is important to note that the generated code was created without any expertise in programming and simulation modeling. The AI application used set the random parameters on its own regarding each of the three scenarios studied. This gives project managers a chance to use advanced computer simulation tools that have been available for a long time but in a much faster time and without the expertise to build such models in specialized simulation software. The code generated in Python:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# General parameters
NUM_SPRINTS = 20
SPRINT_DURATION = 2  # weeks
# Input data for three projects
PROJECTS = {
  "Project 1": {
    "total_sp": 10000,
    "base_velocity": 480,
    "velocity_noise": 30,
    "ac_base": 24000,
    "ac_variation": 1000,
    "bugs_per_sprint": 10,
    "bug_fix_time_mean": 2,
    "bug_fix_time_std": 0.5
  },
  "Project 2": {
    "total_sp": 20000,
    "base_velocity": 420,
    "velocity_noise": 50,
    "ac_base": 130000,
    "ac_variation": 5000,
    "bugs_per_sprint": 25,
    "bug_fix_time_mean": 3.5,
    "bug_fix_time_std": 1.0
  },
  "Project 3": {
```

```
        "total_sp": 15000,
        "base_velocity": 460,
        "velocity_noise": 40,
        "ac_base": 52000,
        "ac_variation": 3000,
        "bugs_per_sprint": 15,
        "bug_fix_time_mean": 2.5,
        "bug_fix_time_std": 0.7
    }
}
# Function to generate sprint data
def generate_project_data(name, params):
    records = []
    ev_cum, pv_cum, ac_cum = 0, 0, 0
    pv = params["total_sp"] / NUM_SPRINTS
    for sprint in range(1, NUM_SPRINTS + 1):
        velocity = max(0, np.random.normal(params["base_velocity"], params["velocity_noise"]))
        ev = min(pv, velocity)
        ac = np.random.normal(params["ac_base"], params["ac_variation"])
        ev_cum += ev
        pv_cum += pv
        ac_cum += ac
        cpi = ev_cum / ac_cum if ac_cum != 0 else 0
        spi = ev_cum / pv_cum if pv_cum != 0 else 0
        bugs = np.random.poisson(params["bugs_per_sprint"])
        fix_time = np.random.normal(params["bug_fix_time_mean"], params["bug_fix_time_std"])
        records.append({
            "Project": name,
            "Sprint": sprint,
            "PV": pv,
            "EV": ev,
            "AC": ac,
            "CPI": cpi,
            "SPI": spi,
            "Velocity": velocity,
            "Defects": bugs,
            "Bug Fix Time": fix_time })
    return pd.DataFrame(records)
# Generate data for all projects
all_data = pd.concat([
    generate_project_data(name, params) for name, params in PROJECTS.items()
], ignore_index=True)
# Set seaborn style
sns.set(style="whitegrid")
# Function to create and save plots
def plot_and_save(metric, ylabel, title, filename, kind="line", marker=False):
    plt.figure(figsize=(10, 6))
    for project in all_data["Project"].unique():
        subset = all_data[all_data["Project"] == project]
        if kind == "line":
            plt.plot(subset["Sprint"], subset[metric], label=project,
                     marker='o' if marker else None)
        elif kind == "bar":
            plt.bar(subset["Sprint"] + 0.25 * (list(PROJECTS.keys()).index(project)-1),
                    subset[metric], width=0.25, label=project)
```

```
        plt.title(title)
        plt.xlabel("Sprint")
        plt.ylabel(ylabel)
        plt.legend()
        plt.tight_layout()
        plt.savefig(filename)
        plt.close()
    # Generate and save all plots
    plot_and_save("CPI", "CPI", "CPI Over Time", "cpi_plot.png", marker=True)
    plot_and_save("SPI", "SPI", "SPI Over Time", "spi_plot.png", marker=True)
    plot_and_save("Velocity", "Velocity", "Velocity Over Time", "velocity_plot.png")
    plot_and_save("Defects", "Number of Defects", "Defects Over Time", "defects_plot.png",
kind="bar")
    plot_and_save("Bug Fix Time", "Fix Time (days)", "Bug Fix Time Over Time", "bug_fix_plot.png",
kind="bar")
    # Boxplot for velocity
    plt.figure(figsize=(8, 6))
    sns.boxplot(data=all_data, x="Project", y="Velocity", hue="Project", palette="pastel",
legend=False)
    plt.title("Velocity Distribution by Project")
    plt.xlabel("Project")
    plt.ylabel("Velocity")
    plt.tight_layout()
    plt.savefig("velocity_boxplot.png")
    plt.close()
```

Finally, according to the given prompt, the results of the simulation are shown in the graphs on the output variables (figure 2). The simulation was carried out in Pycharm 2024.3.11 software.

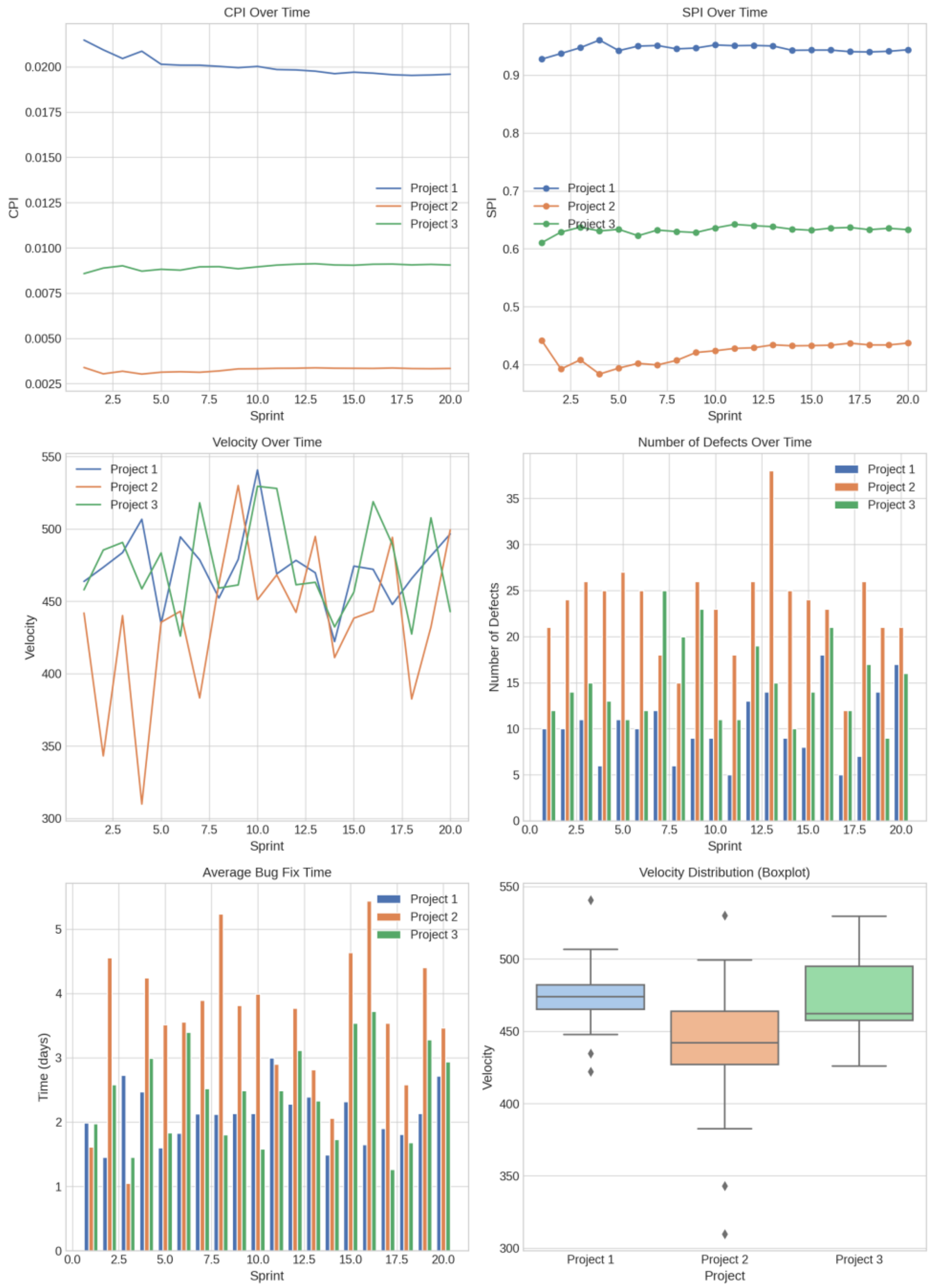

Figure 2. Results from simulation implementation

Based on the presented results, it can be observed that the three simulated project scenarios exhibit significant differences in key performance indicators. These differences reflect the variation in team structures, project complexity, and leadership characteristics assumed in the model. The simulation results, demonstrate how EVM metrics – including CPI, SPI, velocity, number of defects, and average bug fix time – respond to structural and organizational variables. Project 1 showed the highest stability and efficiency: its CPI and SPI values were close to optimal, while the team maintained consistently high velocity and quality (low number of defects and short resolution times). Project 2 produced the weakest results, with low cost and schedule performance, high defect rates, and long bug fix times, indicating major organizational issues and poor team cohesion. Project 3 occupied an intermediate position: initially inefficient, it showed gradual improvements in velocity, defect reduction, and process stability over time. In summary, these results highlight the importance of human factors – such as team cohesion, role clarity, and adaptive capacity – in determining the success of Agile projects. While Earned Value Management provides a quantitative framework for monitoring progress, the quality and reliability of project outcomes depend largely on soft factors shaping team dynamics. The conducted what-if analysis effectively illustrated how relatively small variations in team composition, leadership style, or task complexity can significantly affect key indicators of performance, including cost efficiency, delivery speed, and overall product quality.

5.1 Comparative simulation evaluations

The study validates a simulation model developed using proposed framework (generated Python code for simulation by GPT4) by benchmarking its outputs across three independent projects against two alternative models. The first comparison model was built in ARENA, a discrete-event simulation platform widely recognized for its accuracy in modeling project management processes (Fang & Marle, 2012). The second comparator employed Asana, a popular project management solution recently enhanced with AI features (e.g. AI teammates, smart workflows) that automate task execution and provide intelligent insights. The validation considered averages from 20 sprint iterations for key EVM metrics (CPI, SPI), velocity, defect count, and average bug-fix time. The summary results are presented in table 4.

Table 4. Comparison of average results between analyzed models

Project	Method	CPI	SPI	Velocity	Defects	Avg Bug-Fix Time (days)
1	GPT4	0.0190	0.91	480	12	2.0
	ARENA	0.0188	0.909	478	13	2.1
	Asana AI	0.0176	0.89	455	17	2.7
2	GPT4	0.0028	0.41	340	24	4.5
	ARENA	0.0027	0.408	338	25	4.6
	Asana AI	0.0025	0.395	320	30	5.3
3	GPT4	0.0092	0.63	495	15	3.2
	ARENA	0.0090	0.628	492	16	3.3
	Asana AI	0.0087	0.61	470	20	3.9

Source: own elaboration.

In the ARENA and GPT4 models, discrepancies across CPI/SPI are under 2 %, velocity under 1 %, and defect and fix-time differences are under 10 %, indicating strong alignment. The Asana AI model, by contrast, exhibits larger variances: CPI and SPI lower by 5–10 %, velocity reduced by 5–7 %, defect counts increased by 20–25 %, and bug-fix time increased by similar magnitude.

ARENA's strength lies in its granular modeling of structural and event-driven dynamics, and the simulation model generated by GPT4 in Python code successfully replicates these relationships. Asana AI, while efficient in operational project management, applies abstraction layers and rule-based decision models that oversimplify the simulation, resulting in lower fidelity and larger performance deviations. The alignment between the GPT4 and ARENA simulation models demonstrates that GPT4 can accurately restore both structural integrity and behavioral fidelity of a manual simulation framework.

6. CONCLUSION

Computer simulations and generative AI offer innovative solutions by modeling project scenarios, automating repetitive tasks, and improving resource allocation. The framework proposed in the paper integrates these technologies into project management processes with the objective of enhancing operational efficiency, strengthening intelligent management practices, and mitigating uncertainties. Traditionally, simulation has been used by project managers to optimize decision-making, but its implementation required proficiency in specialized software. This paper demonstrates how Large Language Models can build simulation models for scenario analysis in project management using natural language descriptions. An adaptation of the Earned Value method to Scrum conditions was developed, integrating manufacturing process flows with Agile EVM variable calculations in a simulation model.

The results show that the proposed method simplifies the development of simulation models, making it a valuable tool for analyzing the progress of IT projects managed under Agile-based methodologies. Generative AI and simulation modeling emerge as complementary techniques that can deliver the actionable insights needed for intelligent management, while intuitive visualizations further support effective decision-making.

The concept provides a strong foundation for further research into the integration potential of generative AI in project management within complex organizational environments. Future work should explore extending this approach with modules for advanced risk analysis, cost estimation, and predictive scenario forecasting in multi-team projects. Empirical implementations could enable systematic evaluation of LLM-based tools on process efficiency and deliverable quality within Agile development frameworks.

Funding: Co-financed by the Polish Minister of Science under the "Regional Excellence Initiative".

Regionalna Inicjatywa Doskonałości

Ministry of Science and Higher Education Republic of Poland

REFERENCES

Acebes, F., Pajares, J., Galán, J. M., & Lopez-Paredes, A. (2014). A new approach for project control under uncertainty, going back to the basics. *International Journal of Project Management, 32*, 423–434.

AlShannaq, F. B., Shehab, M. M., Al-Assaf, A. H., Alhenawi, E., & Awawdeh, S. (2024). An exploration into the mechanisms and evolution of GPT models. In *Impacts of generative AI on the future of research and education* (477).

Aramali, V., Gibson, G. E., Jr., El Asmar, M., & Cho, N. (2022). Novel earned value management system maturity framework and its relation to project performance. *Journal of Construction Engineering and Management, 148*(4).

Auth, G., JokischPavel, O., & Dürk, C. (2019). Revisiting automated project management in the digital age: A survey of AI approaches. *Online Journal of Applied Knowledge Management, 7*(1), 27–39.

Banks, J., Carson, J. S., & Nelson, B. L. (2005). *Discrete-event system simulation* (4th ed.). Pearson Prentice Hall.

Beck, K. (2005). *Extreme programming explained: Embrace change* (2nd ed.). Addison-Wesley.

Boehm, B., & Turner, R. (2004). *Balancing agility and discipline: A guide for the perplexed*. Addison-Wesley.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., & Kaplan, J. (2021). Evaluating large language models trained on code. *arXiv preprint* arXiv:2107.03374.

Cho, N., El Asmar, M., Gibson, G. E., Jr., & Aramali, V. (2020). Earned value management system (EVMS) reliability: A review of existing EVMS literature. In W. D. Grau, G. Parrish, & M. Chong (Eds.), *Construction research congress 2020: Infrastructure systems and sustainability* (pp. 1004–1013). American Society of Civil Engineers.

Choudhary, T. (2025). Political bias in large language models: A comparative analysis of ChatGPT-4, Perplexity, Google Gemini, and Claude. *IEEE Access, 13*, 11341–11379.

Cockburn, A. (2004). *Crystal clear: A human-powered methodology for small teams*. Addison-Wesley.

Crawford, T., Duong, S., Fueston, R., Lawani, A., Owoade, S., Uzoka, A., Yazdinejad, A., et al. (2023). AI in software engineering: A survey on project management applications. *arXiv preprint* arXiv:2307.15224.

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software, 85*(6), 1213–1221.

Durth, S., Hancock, B., Maor, D., & Sukharevsky, A. (2023). *The organization of the future: Enabled by gen AI, driven by people*. McKinsey & Company.

Fang, C., & Marle, F. (2012). A simulation-based risk network model for decision support in project risk management. *Decision Support Systems, 52*(3), 635–644.

Government Accountability Office. (2021). *NASA: Earned value management implementation across major spaceflight projects is uneven, and factors influencing implementation are under-addressed (GAO-13-22)*.

Hanne, T., & Nickel, S. (2005). A multiobjective evolutionary algorithm for scheduling and inspection planning in software development projects. *European Journal of Operational Research, 167*(3), 663–678.

Highsmith, J. (2009). *Adaptive software development: A collaborative approach to managing complex systems*. Addison-Wesley.

Hofmann, P., Jöhnk, J., Protschky, D., & Urbach, N. (2020). Developing purposeful AI use cases: A structured method and its application in project management. In *Wirtschaftsinformatik (Zentrale Tracks)* (pp. 33–49).

Jackson, I., Saenz, M. J., & Ivanov, D. (2023). From natural language to simulations: Applying AI to automate simulation modelling of logistics systems. *International Journal of Production Research, 62*(4), 1434–1457.

Lindsjørn, Y., Dingsøyr, T., Bergersen, G. R., & Sjøberg, D. I. (2016). Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software, 122*, 274–286.

Mossavar-Rahmani, F., & Zohuri, B. (2024). ChatGPT and beyond: The next generation of AI evolution (A communication). *Journal of Energy and Power Engineering, 18*, 146–154.

Naeni, L. M., & Salehipour, A. (2011). Evaluating fuzzy earned value indices and estimates by applying alpha cuts. *Expert Systems with Applications, 38*, 8193–8198.

National Defense Industrial Association. (2018). *Earned value management systems EIA-748-D intent guide*.

Nguyen, D. A., Cabrero-Daniel, B., Przybylek, A., Arora, C., Khanna, D., Herda, T., Rafiq, U., Melegati, J., Guerra, E., Kemell, K. K., Saari, M., Zhang, Z., Le, H., Quan, T., & Abrahamsson, P. (2023). Generative artificial intelligence for software engineering: A research agenda. *SSRN.*

Palmer, S. R., & Felsing, J. M. (2002). *A practical guide to feature-driven development*. Pearson.

Pikh, I., Senkivskyy, V., Sikora, L., Lysa, N., & Kudriashova, A. (2025). Automated system for evaluating alternatives for developing innovative IT projects. *Applied Sciences, 15*(3), 1167.

Pricopoaia, O., Busila, A. V., Cristache, N., Susanu, I., & Matis, C. (2024). Challenges for entrepreneurial innovation: Startups as tools for a better knowledge-based economy. *International Entrepreneurship and Management Journal, 20*, 969–1010.

Project Management Institute. (2018). *A guide to the project management body of knowledge (PMBOK guide)* (6th ed.).

Ratushny, R., Tryhuba, A., Bashynsky, O., & Ptashnyk, V. (2019). Development and usage of a computer model of evaluating the scenarios of projects for the creation of fire fighting systems of rural communities. In *XIth International Scientific and Practical Conference on Electronics and Information Technologies (ELIT)* (pp. 34–39). IEEE.

Rodríguez, D., Ruiz, M., Riquelme, J. C., & Harrison, R. (2011). Multiobjective simulation optimisation in software project management. In *Proceedings of the 13th annual conference on genetic and evolutionary computation* (pp. 1883–1890).

Schwaber, K., & Sutherland, J. (2020). *The Scrum guide: The definitive guide to Scrum: The rules of the game*. Scrum.org.

Sravanthi, J., Sobti, R., Semwal, A., Shravan, M., Al-Hilali, A. A., & Alazzam, M. B. (2023). AI-assisted resource allocation in project management. In *3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (pp. 70–74). IEEE.

Stapleton, J. (1997). *DSDM: The method in practice*. Addison-Wesley.

Szyjewski, Z., & Muszyńska, K. (2013). *Zarządzanie projektami i modelowanie procesów*. Polskie Towarzystwo Informatyczne.

Uppalapati, V. K., & Nag, D. S. (2024). A comparative analysis of AI models in complex medical decision-making scenarios: Evaluating ChatGPT, Claude AI, Bard, and Perplexity. *Cureus, 16*(1).

Wooley, A., Silva, D. F., & Bitencourt, J. (2023). When is a simulation a digital twin? A systematic literature review. *Manufacturing Letters, 35*, 940–951.